MS Appeal Brief-Patents
PATENT
1533-1005

IN THE U.S. PATENT AND TRADEMARK OFFICE BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

In re application of	Appeal No.
Gunnar NORDMARK et al.	Conf. 6777
Application No. 10/510,167	Group 2446
Filed: July 18, 2005	Examiner Scott M. Sciacca

Title: METHOD AND APPARATUS FOR PROCESSING DATA

APPEAL BRIEF

MAY IT PLEASE YOUR HONORS:

(i) **Real Party in Interest**

The real party of interest in this appeal is Xelerated AB, Olof Palmes gata 29, SE-111 22 Stockholm, Sweden.

(ii) **Related Appeals and Interferences**

None.

(iii) **Status of Claims**

Claim 1, 5-9 and 11-22 are pending and stand rejected. Claims 2-4 and 10 were previously cancelled. This appeal is taken from the final rejection of claims 1, 5-9 and 11-22.

(iv) **Status of Amendments**

All Amendments have been entered. No amendment has been filed subsequent to the final rejection.

(v) **Summary of Claimed Subject Matter**

Independent claim 1 is directed to a method for pipelined processing of a data packet in at least two stages where an intermediate packet is generated with a dummy header and/or tail.

Claim 1 recites a method of pipelined processing of a data packet (100,315) (See Figs. 3A and 3b page 6, LINES 1-24) in a processing means (700) (See Fig. 7, page 14, lines 13-23) comprising at least two processing stages (205) (See Fig. 2 and page 4, line 16 through page 5, line 31), said data packet (100) (See Fig. 3a) containing information, said method comprising: generating an intermediate data packet (315) (See Fig. 3b, page 6, lines 1-24) by adding at least one of a dummy header (305) (See Fig. 3b) and dummy tail (310) (See Fig. 3b) to said data packet (100), said dummy header (305) and dummy tail (310) (See Fig. 3b) being capable of storing information of a communication system (page 6, lines 1-24); associating (510) (See Fig. 5) information reference (320; 325,330) to said intermediate data packet (315) (See Figs. 3a-3d and 5 and page 9, lines 11-18), said information reference (320; 325,330) comprising information relating to the length and position of the information of said data packet (100) contained in said intermediate data packet (315) (See Figs. 3a-3d and page 9, lines 11-18); storing said information reference (320, 325, 330) in additional register (230) (See Fig. 2 and page 9, line 15); processing (520) (See

Fig. 5) said intermediate data packet (315) in a processing stage (205) (see Fig. 2; and page 9, line 18 and 19); and said processing (520) (See Fig. 5) of said intermediate data packet (315) (See Figs. 3b and 3c) results in a change of the length of said information of said data packet (100) contained in said intermediate data packet (315) (See page 9, lines 19-21), whereby said information reference (320; 325, 330) (See Figs 3a-3d) is altered (530) in order for said information reference (320; 325, 330) to reflect said change (See page 9, lines , lines 21-26), wherein the change in the length of said information of said data packet (100) (See Fig. 3a-3d) comprises: adding to the length a value representing a length of a portion of the at least one of the dummy header (305) and dummy tail (310) of said intermediate data packet (315) (page 9, lines 9-11), the portion storing information of the communications system (See Fig. 3a-3d), and subtracting from the length a value representing a portion of said intermediate data packet representing empty information after said processing (See Fig. 3d and page 7, lines 11-20), the method further comprising the step of determining, upon said intermediate data packet (315) exiting the last of said at least two processing stages (205b,205c), (540) (See Fig. 5) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous (See Fig. 5, page 9, lines 31 and 32), then removing (545) (See Fig. 5) said superfluous bits from at least one of the header end and the tail

end of said intermediate data packet (315) (See Fig. 5, page 10, lines 1-4).

Independent claim 9 is directed to a processing means for pipelined processing of a data packet in at least two stages where an intermediate packet is generated with a dummy header and/or tail.

Claim 9 recites a processing means (See Fig. 7, 700, page 14, lines 13-23) for pipelined processing of a data packet (100) (See Figs. 3A and 3b page 6, lines 1-24), said processing means comprising at least two processing stages (205a, 205b, 205c) (See Fig. 2) comprising a logic unit (210) and a register (220) (See page 4, line 16 through page 5, line 5) for storing at least part of said data packet (100)(See Fig. 3a-3d), said processing means comprising: a receiver (705) (See Fig. 7, page 14, line 16) is adapted to receive said data packet (100) and to generate an intermediate data packet (315) by adding at least one of a dummy header (305) (See Fig. 3b) and a dummy tail (310) (See Fig. 3b) to said data packet (100) (see page 9, lines 9-11), said dummy header (305) and dummy tail (310) (See Fig. 3b) being capable of storing information of a communications system (See page 14, line 30 through page 15, line 4); at least one register (230) (See Fig. 2) for storing information reference (320) (See Figs. 3a-3d) associated with said intermediate data packet (315) is accessible to said logic unit (210), said information reference (320;325,330) comprising information relating to the

length and position of the information of the data packet (100) contained in said intermediate data packet (315) (See page 1, lines 20-24); at least one of at said at least one logic units (210) is adapted to operate upon said information reference (320) (See page 4, lines 18 and 19); and a processing stage (205) for processing said intermediate data packet (315) (See Fig. 2 and page 4, lines 16-19 and 29-32), wherein the length of the information of the data packet (100) contained in said intermediate data packet (315) changes upon processing said intermediate data packet (315) in said processing stage (205) (See page 9, lines 9-11 and 15-19), whereby said information reference (320,325,330) is altered (530) in order for said information reference (320,325,330) to reflect said change (See page 9, lines 21-26), wherein the change in the length of said information of said data packet (110) comprises: adding to the length a value representing a length of the portion of the at least one of the dummy header (305) and dummy tail (310) of said intermediate data packet (315) (page 9, lines 9-11), the portion storing information of the communications system (See Fig. 3a-3d), and subtracting from the length a value representing a portion of said intermediate data packet (315) comprising empty information after said processing (See Fig. 3d and page 7, lines 11-20), the processing means further comprising: a last processing stage (205b, 205c) configured to determine, upon said intermediate data packet (315) exiting the last of the at least

two processing stages (205b, 205c), (540) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous (See Fig. 5, page 9, lines 31 and 32), then removing (545) said superfluous bits from the at least one of the header end and the tail end of said intermediate data packet (315) (See Fig. 5, page 10, lines 1-4).

Independent claim 19 is directed to a pipelined processor for processing data packet in at least two stages where an intermediate packet is generated with a dummy header and/or tail.

Claim 19 recites a pipelined processor for processing a data packet (100) (See Fig. 7, 700, page 14, lines 13-23), comprising: a register (100) for storing at least part of the data packet (100) (See Fig. 3a-3d); at least one additional register (230) for storing an information reference (320) for association with an intermediate data packet (315) (See Fig. 2 and page 9, line 15); a logic unit (210) (See Fig. 2 and 6a-6d) performing the steps of: receiving the data packet (100) (See Fig. 5, 500, page 9, lines 6-8); generating the intermediate data packet (315) by adding at least one of a dummy header (305) and dummy tail (310) (See Figs. 3a-3d) to the data packet (100) (see page 9, lines 9-11); associating (510) information reference (320; 325,330) to the intermediate data packet (315), the information reference (320; 325,330) comprising information

relating to the length and position of the information of the data packet (100) contained in the intermediate data packet (315) (See Figs. 3a-3d and 5 and page 9, lines 11-18); storing the information reference (320, 325, 330) in the at least one additional register (230) (See Fig. 2 and page 9, line 15); processing (520) the intermediate data packet (315) in a processing stage (205) (See page 9, lines 19-21); and said processing (520) of the intermediate data packet (315) results in a change of the length of said information of the data packet (100) contained in the intermediate data packet (315), then altering (530) the information reference (320; 325,330) in order for the information reference (320; 325, 330) to reflect the change (See page 9, lines , lines 21-26), wherein the change in the length of said information of said data packet (100) comprises: adding to the length a value representing a length of a portion of the at least one of the dummy header (305) and dummy tail (310) of said intermediate data packet (315), the portion storing information of the communications system (page 9, lines 9-11), and subtracting from the length a value representing a portion of said intermediate data packet (315) comprising empty information after said processing (See Fig. 3d and page 7, lines 11-20), the logic unit (210) further performing the step of: determining, upon said intermediate data packet (315) exiting the last of at least two processing stage (205b, 205c), (540) whether any bits of at least one of the header end and the tail end of

said intermediate data packet (315) are superfluous (See Fig. 5, page 9, lines 31 and 32), then removing (545) said superfluous bits from at least one of the header end and the tail end of said intermediate data packet (315) (See Fig. 5, page 10, lines 1-4).

(vi) **Grounds of Rejection to be Reviewed on Appeal**

The first issue on appeal is whether claims 1, 5-9, 11-12 and 15-22 would have been obvious, in the meaning of 35 USC §103(a), based on Sonksen, U.S. Patent Publication No. 2003/0046429 in view of Kawarai, U.S. Patent Publication No. 2002/0122424 in view of Farinacci, U.S. Patent No. 7,016,351 in view of Hultsch WO 99/60708.

The second issue on appeal is whether claim 13 would have been obvious, in the meaning of 35 USC §103(a), based on Sonksen, U.S. Patent Publication No. 2003/0046429 in view of Kawarai, U.S. Patent Publication No. 2002/0122424 in view of Farinacci, U.S. Patent No. 7,016,351 in view of Hultsch WO 99/60708 in further view of Lee, U.S. Patent No. 6,996,117.

The third issue on appeal is whether claim 14 would have been obvious, in the meaning of 35 USC §103(a), based on Sonksen, U.S. Patent Publication No. 2003/0046429 in view of Kawarai, U.S. Patent Publication No. 2002/0122424 in view of Farinacci, U.S. Patent No. 7,016,351 in view of Hultsch WO 99/60708 in further view of Song, U.S. Patent No. 5,818,894.

(vii) **Arguments**

(1) Arguments Concerning the First Ground of Rejection, Claims 1, 5-9, 11-12 and 15-22 would not have been obvious based on Sonksen, U.S. Patent Publication No. 2003/0046429 in view of Kawarai, U.S. Patent Publication No. 2002/0122424 in view of Farinacci, U.S. Patent No. 7,016,351 in view of Hultsch WO 99/60708.

Sonksen discusses a packet processing apparatus implemented in a plurality of pipeline stages or a group of stages configured for a particular operation.

Kawarai discusses an interface device that is used to accommodate packets from a high-speed line efficiently and to reduce a processing load on a back stage caused by routing control.

Farinacci discusses placing multicast delivery tree information in the header of an encapsulated multicast packet, thereby relieving the intermediate routers from maintaining any state information about the multicast groups.

Hultsch discusses the removal of filler data from a data stream with a constant data rate and the reformatting of the useful data of the constant stream into packetized data. The Applicants note that the corresponding US national stage of WO 99/60708 may be found in U.S. Patent No. 7,415,037.

Claim 1

In the Amendment filed May 11, 2009 the Applicants amended independent claim 1 to further recite "determining, upon said intermediate data packet (315) exiting the last of said at least two processing stages (205b,205c), (540) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous, then removing (545) said superfluous bits from at least one of the header end and the tail end of said intermediate data packet (315)."

On page 6 of the final Office Action dated August 20, 2009, the Office asserts "Sonksen further teaches upon said intermediate data packet exiting the last of said at least one processing stages removing data from said intermediate packet." (See Sonksen, Abstract and Fig. 5a)

Further, on page 6 of the final Office Action the Office acknowledges that "Sonksen does not explicitly teach the removal being based upon a determination of whether bits of said intermediate data packet are superfluous," but asserts that the Abstract of Hultsch discloses such a feature.

However, the Applicants respectfully submit that the combination of the references do not teach the removal of superfluous bits from headers and/or tails of a packet.

It is noted that the rejection never mentions the removal of superfluous bits from the header or the tail. It would appear that the Office implicitly includes the header and

tail as part of the payload of the packet and therefore indistinguishable from the payload. The Applicants respectfully disagree with this interpretation. It is well know in the art that a header defines information about the packet such as its source IP address, destination IP address, Time To Live, checksum information, etc…, (See Fig. 9B of Sonksen) while the payload is the data to be transmitted. In a classic sense routers and other network equipment need not know anything about the payload of a packet, while the header is relied on for the function of network equipment.

The Applicants acknowledge that Sonksen states in ¶ 0013 "[i]n one embodiment the a method and apparatus is configured to dynamically supplement, modify, or remove data contained in a packet. The term packet includes the header or tag information of a packet in addition to any user data associated with packet." However, this can not be interpreted as removing bits from header. In well know terms a packet always contains a header and payload however, the removal of bits from the header will have profound changes on the function of network equipment while the removal from the payload will have none.

Further, Sonksen ¶ 0068 states

> As shown, FIG. 5A is generally similar to FIG. 3 and illustrates a dynamic processing module and a static processing module. The dynamic processing module 502 and the static processing module 504 are part of the processing pipeline. ***The dynamic processing module 502 comprises a module that may be configured to generate or modify a tag or other portion of a packet header,***

> such as a tag that may be attached to a portion of a packet to aid in packet processing or routing. The static processing module comprises a module dedicated to ***modifying or updating the TTL, TOS, error control portions, or any other portion of the packet data to reflect changes to the packet***. Either of the modules 502, 504 may comprise any configuration of hardware, firmware, logic, and/or memory configured to achieve the processes described herein. The static processing module may be configured to support modification of packet header fields for IPv4, IPv6, or other data types. [Emphasis added]

Thus, Sonksen clearly discusses generating or modifying the header of a packet, but remains silent as to removing any superfluous bits of the header.

The only portion on Sonksen that explicitly discusses removing data is ¶ 0094 which states in part "[i]n this example method the dynamic processing module generates, modifies, or removes a tag."

Thus, Sonksen does not disclose the removal of superfluous bits from a header.

Further, the Applicants acknowledge that Hultsch discloses the removal of filler bits from a stream. The filler bits of Hultsch are not found in a header but are added to the payload to a keep a constant data rate via the circuit-switched connection. They are in fact bit stuffing of a payload. This is made clear in col. 3, lines 14-24 which state:

> The compressed data is, by way of example, ***compressed video data*** which is generated by a video coder as a data stream DS1 with a constant data rate DRk--for example 64 kbit/s or 2*64 kbit/s--and is transmitted via at least one circuit-switched connection of the fixed network ISDN. In this case, ***the constant data rate DRk is generated by the addition of filling data F***

> ***to useful data N (bit stuffing).*** The useful data N contains the data which is necessary for the video conference, is generated by a communications terminal of the line-connected fixed network ISDN and is transported in real time to a communications terminal of the mobile communications network UMTS. [Emphasis added]

The Applicants respectfully submit that Sonksen and Hultsch, alone or in combination, fail to disclose "removing (545) said superfluous bits from at least one of the header end and the tail end of said intermediate data packet," as in claim 1.

Additionally, it is respectfully submitted that if arguendo, Sonksen disclosed removing bits from a header and/or tail, one of ordinary skill in the art would not have looked to Hultsch as it is not dealing with removing superfluous bits from packetized data, but a constant stream.

Further, it is submitted that the Office does not assert and the Applicants have not found that Kawarai and Farinacci provide any further disclosure to render the claims obvious as to the features discussed above.

Therefore, Sonksen, Kawarai, Farinacci and Hultsch, taken separately or in combination, fail to render obvious the features of claim 1 or the claims dependent therefrom.

Claim 9

In the Office Action, the Office asserts on page 12 "Sonksen does not explicitly teach the removal being based on a

determination of whether any bits of said intermediate data packet are superfluous. However, Hultsch discloses removing superfluous data from a packet ('removing the superfluous filler data received via the circuit-switched connection in the data stream (OS1) with the constant data rate and by reformatting the useful data for the data stream with the variable data rate and sending it via a packet-orientated connection' - See Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include removing superfluous bits from said intermediate data packet for the same reasons as those given with respect to Claim 1."

It is noted that the rejection never mentions the removal of superfluous bits from the header or the tail. It would appear that the Office implicitly includes the header and tail as part of the payload of the packet and therefore indistinguishable from the payload. The Applicants respectfully disagree with this interpretation. It is well know in the art that a header defines information about the packet such as its source IP address, destination IP address, Time To Live, checksum information, etc…, (See Fig. 9B of Sonksen) while the payload is the data to be transmitted. In a classic sense routers and other network equipment need not know anything about the payload of a packet, while the header is relied on for the function of network equipment.

The Applicants acknowledge that Sonksen states in ¶

0013 “[i]n one embodiment the a method and apparatus is configured to dynamically supplement, modify, or remove data contained in a packet. The term packet includes the header or tag information of a packet in addition to any user data associated with packet.” However, this can not be interpreted as removing bits from header. In well know terms a packet always contains a header and payload however, the removal of bits from the header will have profound changes on the function of network equipment while the removal from the payload will have none.

However as discussed above with regards to claim 1, Sonksen discusses generating or modifying the header of a packet, but remains silent as to removing any superfluous bits of the header.

The only portion on Sonksen that explicitly discusses removing data is ¶ 0094 which states in part “[i]n this example method the dynamic processing module generates, modifies, or removes a tag.”

Thus, Sonksen does not disclose the removal of superfluous bits from a header.

Further, the Applicants acknowledge that Hultsch discloses the removal of filler bits from a stream. The filler bits of Hultsch are not found in a header but are added to the payload to a keep a constant data rate via the circuit-switched connection. They are in fact bit stuffing of a payload. This is made clear in col. 3, lines 14-24 as quoted above.

The Applicants respectfully submit that Sonksen and Hultsch, alone or in combination, fail to disclose "a last processing stage (205b, 205c) configured to determine, upon said intermediate data packet (315) exiting the last of the at least two processing stages (205b, 205c), (540) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous, then removing (545) said superfluous bits from the at least one of the header end and the tail end of said intermediate data packet (315)," as in claim 9.

Additionally, it is respectfully submitted that if arguendo, Sonksen disclosed removing bits from a header and/or tail, one of ordinary skill in the art would not have looked to Hultsch as it is not dealing with removing superfluous bits from packetized data, but a constant stream.

Further, it is submitted that the Office does not assert and the Applicants have not found that Kawarai and Farinacci provide any further disclosure to render the claims obvious as to the features discussed above.

Therefore, Sonksen, Kawarai, Farinacci and Hultsch, taken separately or in combination, fail to render obvious the features of claim 9 or the claims dependent therefrom.

Claim 19

On page 18 of the Office Action, it is asserted that "Sonksen does not explicitly teach the removal being based on a

determination of whether any bits of said intermediate data packet are superfluous. However, Hultsch discloses removing superfluous data from a packet ('removing the superfluous filler data received via the circuit-switched connection in the data stream (OS1) with the constant data rate and by reformatting the useful data for the data stream with the variable data rate and sending it via a packet-orientated connection' - See Abstract). It would have been obvious to one of ordinary skill in the art at the time the invention was made to modify Sonksen to include removing superfluous bits from said intermediate data packet for the same reasons as those given with respect to Claim 1."

The Applicants acknowledge that Sonksen states in ¶ 0013 "[i]n one embodiment the a method and apparatus is configured to dynamically supplement, modify, or remove data contained in a packet. The term packet includes the header or tag information of a packet in addition to any user data associated with packet." However, this can not be interpreted as removing bits from header. In well know terms a packet always contains a header and payload however, the removal of bits from the header will have profound changes on the function of network equipment while the removal from the payload will have none.

However as discussed above with regards to claim 1, Sonksen discusses generating or modifying the header of a packet, but remains silent as to removing any superfluous bits of the header.

The only portion on Sonksen that explicitly discusses removing data is ¶ 0094 which states in part "[i]n this example method the dynamic processing module generates, modifies, or removes a tag."

Thus, Sonksen does not disclose the removal of superfluous bits from a header.

Further, the Applicants acknowledge that Hultsch discloses the removal of filler bits from a stream. The filler bits of Hultsch are not found in a header but are added to the payload to a keep a constant data rate via the circuit-switched connection. They are in fact bit stuffing of a payload. This is made clear in col. 3, lines 14-24 as quoted above.

The Applicants respectfully submit that Sonksen and Hultsch, alone or in combination, fail to disclose "determining, upon said intermediate data packet (315) exiting the last of at least two processing stage (205b, 205c), (540) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous, then removing (545) said superfluous bits from at least one of the header end and the tail end of said intermediate data packet (315)" as in claim 19.

Additionally, it is respectfully submitted that if arguendo, Sonksen disclosed removing bits from a header and/or tail, one of ordinary skill in the art would not have looked to Hultsch as it is not dealing with removing superfluous bits from

packetized data, but a constant stream.

Further, it is submitted that the Office does not assert and the Applicants have not found that Kawarai and Farinacci provide any further disclosure to render the claims obvious as to the features discussed above.

Therefore, Sonksen, Kawarai, Farinacci and Hultsch, taken separately or in combination, fail to render obvious the features of claim 19 or the claims dependent therefrom.

(2) Arguments Concerning the Second Ground of Rejection, claim 13 would not have been obvious based on Sonksen, U.S. Patent Publication No. 2003/0046429 in view of Kawarai, U.S. Patent Publication No. 2002/0122424 in view of Farinacci, U.S. Patent No. 7,016,351 in view of Hultsch WO 99/60708 in further view of Lee, U.S. Patent No. 6,996,117.

Claim 13 is dependent from allowable base claims as discussed above. As such, the combination Sonksen, Kawarai, Farinacci, Hultsch and Lee fails to render claim 13 obvious.

(3) Arguments Concerning the Third Ground of Rejection, claim 14 would not have been obvious based on Sonksen, U.S. Patent Publication No. 2003/0046429 in view of Kawarai, U.S. Patent Publication No. 2002/0122424 in view of Farinacci, U.S. Patent No. 7,016,351 in view of Hultsch WO 99/60708 in further view of Song, U.S. Patent No. 5,818,894.

Claim 14 is dependent from allowable base claims as discussed above. As such, the combination Sonksen, Kawarai, Farinacci, Hultsch and Song fails to render claim 14 obvious.

Conclusion

Appellants respectfully urge that the rejections on appeal should not be maintained, and respectfully requests that these rejections be reversed.

Respectfully submitted,

YOUNG & THOMPSON

/James J. Livingston, Jr./

James J. Livingston, Jr.
Reg. No. 55,394
209 Madison Street, Suite 500
Alexandria, VA 22314
Telephone (703) 521-2297
Telefax (703) 685-0573
(703) 979-4709

JJL/fb

May 27, 2010

(viii) Claims Appendix

1. A method of pipelined processing of a data packet (100,315) in a processing means (700) comprising at least two processing stages (205), said data packet (100) containing information, said method comprising:

generating an intermediate data packet (315) by adding at least one of a dummy header (305) and dummy tail (310) to said data packet (100) , said dummy header (305) and dummy tail (310) being capable of storing information of a communication system;

associating (510) information reference (320; 325,330) to said intermediate data packet (315), said information reference (320; 325,330) comprising information relating to the length and position of the information of said data packet (100) contained in said intermediate data packet (315);

storing said information reference (320, 325, 330) in additional register (230);

processing (520) said intermediate data packet (315) in a processing stage (205); and

said processing (520) of said intermediate data packet (315) results in a change of the length of said information of said data packet (100) contained in said intermediate data packet (315), whereby said information reference (320; 325, 330) is altered (530) in order for said information reference (320; 325, 330) to reflect said change,

wherein the change in the length of said information of said data packet (100) comprises:

adding to the length a value representing a length of a portion of the at least one of the dummy header (305) and dummy tail (310) of said intermediate data packet (315), the portion storing information of the communications system, and

subtracting from the length a value representing a portion of said intermediate data packet representing empty information after said processing, the method further comprising the step of

determining, upon said intermediate data packet (315) exiting the last of said at least two processing stages (205b,205c), (540) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous, then removing (545) said superfluous bits from at least one of the header end and the tail end of said intermediate data packet (315).

5. The method of claim 1, the method further comprising the steps of:

removing, upon said intermediate data packet (315) exiting the last of said at least one processing stages, at least one bit from said intermediate data packet (315).

6. The method of claim 1, wherein said information reference (320) is included in additional information (225) associated with said intermediate data packet (315).

7. The method of claim 1, wherein prior to said step of processing (520) said intermediate data packet (315), said information reference (320) is stored in at least one register (230) accessible to the processing stage (205) performing said processing (520).

8. The method of claim 1, wherein said information reference comprises a length value (325) and an offset value (330), said length value (325) representing the length of the information contained in said intermediate data packet (315) and said offset value (330) indicating the position in said intermediate data packet (315) of the information.

9. A processing means for pipelined processing of a data packet (100), said processing means comprising at least two processing stages (205a, 205b, 205c) comprising a logic unit (210) and a register (220) for storing at least part of said data packet (100), said processing means comprising:

a receiver (705) is adapted to receive said data packet (100) and to generate an intermediate data packet (315) by adding at least one of a dummy header (305) and a dummy tail (310) to said data packet (100), said dummy header (305) and dummy tail (310) being capable of storing information of a communications system;

at least one register (230) for storing information reference (320) associated with said intermediate data packet (315) is accessible to said logic unit (210), said information reference (320;325,330) comprising information relating to the length and position of the information of the data packet (100) contained in said intermediate data packet (315);

at least one of at said at least one logic units (210) is adapted to operate upon said information reference (320); and

a processing stage (205) for processing said intermediate data packet (315), wherein the length of the information of the data packet (100) contained in said intermediate data packet (315) changes upon processing said intermediate data packet (315) in said processing stage (205), whereby said information reference (320,325,330) is altered (530) in order for said information reference (320,325,330) to reflect said change,

wherein the change in the length of said information of said data packet (110) comprises:

adding to the length a value representing a length of the portion of the at least one of the dummy header (305) and dummy tail (310) of said intermediate data packet (315), the portion storing information of the communications system, and

subtracting from the length a value representing a portion of said intermediate data packet (315) comprising empty information after said processing, the processing means further comprising:

a last processing stage (205b, 205c) configured to determine, upon said intermediate data packet (315) exiting the last of the at least two processing stages (205b, 205c), (540) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous, then removing (545) said superfluous bits from the at least one of the header end and the tail end of said intermediate data packet (315).

11. The processing means of claim 9, wherein said receiver (715) for adding comprises a buffer (720) and a shifter (725).

12. The processing means of claim 9, the processing means further comprising means (730) for removing at least one bit from said intermediate data packet (315).

13. The processing means of claim 9, wherein means (730) for removing comprises a shifter (735) and a buffer (740).

14. The processing means of claim 11, wherein said shifter (725,735) is a barrel shifter.

15. The processing means (700) of claim 9, wherein said at least one register (230) for storing information reference (230) is located in said processing stage (205).

16. The processing means (700) of claim 9, wherein said at least one register (230) for storing information reference comprises one register (230) for storing a length value (325) and another register (230) for storing an offset value (330).

17. An integrated circuit, characterised by a processing means (700) according to claim 9.

18. A computer unit characterised by an integrated circuit according to claim 9.

19. A pipelined processor for processing a data packet (100), comprising:

a register (100) for storing at least part of the data packet (100);

at least one additional register (230) for storing an information reference (320) for association with an intermediate data packet (315) ;

a logic unit (210) performing the steps of:

receiving the data packet (100);

generating the intermediate data packet (315) by adding at least one of a dummy header (305) and dummy tail (310) to the data packet (100);

associating (510) information reference (320; 325,330) to the intermediate data packet (315), the information reference (320; 325,330) comprising information relating to the length and position of the information of the data packet (100) contained in the intermediate data packet (315);

storing the information reference (320, 325, 330) in the at least one additional register (230);

processing (520) the intermediate data packet (315) in a processing stage (205); and

said processing (520) of the intermediate data packet (315) results in a change of the length of said information of the data packet (100) contained in the intermediate data packet (315), then altering (530) the information reference (320; 325,330) in order for the information reference (320; 325, 330) to reflect the change,

wherein the change in the length of said information of said data packet (100) comprises:

adding to the length a value representing a length of a portion of the at least one of the dummy header (305) and dummy tail (310) of said intermediate data packet (315), the portion storing information of the communications system, and

subtracting from the length a value representing a portion of said intermediate data packet (315) comprising empty information after said processing, the logic unit (210) further performing the step of:

determining, upon said intermediate data packet (315) exiting the last of at least two processing stage (205b, 205c), (540) whether any bits of at least one of the header end and the tail end of said intermediate data packet (315) are superfluous, then removing (545) said superfluous bits from at least one of the header end and the tail end of said intermediate data packet (315).

20. The method of claim 1, wherein said processing (520) of said intermediate data packet (315) further results in a change of the position of said information of said data packet (100) contained in said intermediate data packet (315), whereby said information reference (320; 325,330) is altered (530) in order for said information reference (320; 325, 330) to reflect said change, and wherein the change in the position of said information of said data packet (100) comprises at least one of:

subtracting from the position a value representing a length of the portion of the dummy header (305) of said intermediate data packet (315) containing the information of the communications system, and

adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing.

21. The processing means of claim 9, wherein the processing stage (205) for processing said intermediate data packet (315), is configured to process said intermediate data packet (315) resulting in a change of the position of said information of said data packet (100) contained in said intermediate data packet (315), whereby said information reference (320; 325,330) is altered (530) in order for said information reference (320; 325, 330) to reflect said change, and wherein the change in the position of said information of said data packet (100) comprises at least one of:

subtracting from the position a value representing a length of the portion of the dummy header (305) of said intermediate data packet (315) containing the information of the communications system, and

adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing.

22. The pipelined processor of claim 19, wherein the processing stage (205) for processing said intermediate data packet (315), is configured to process said intermediate data packet (315) resulting in a change of the position of said information of said data packet (100) contained in said intermediate data packet (315), whereby said information reference (320; 325,330) is altered (530) in order for said information reference (320; 325, 330) to reflect said change, and wherein the change in the position of said information of said data packet (100) comprises at least one of:

subtracting from the position a value representing a length of the portion of the dummy header (305) of said intermediate data packet (315) containing the information of the communications system, and

adding to the position a value representing a portion of said intermediate data packet representing empty information after said processing.

(ix) **Evidence Appendix**

None.

(x) **Related Proceedings Appendix**

None.